[current adjustable part I know]

rn.seed

np.random.seed

tf.set\_random\_seed

PYTHONHASHSEED

not using GPU # necessary

config session # necessary

[weight initialization]

way1: set kernel\_initializer seed in model builing

way2: set tf.set\_random(seed) before model building

[training]

* multi-threading error:

When using GPU or CPU which has multi-thread capability, the calculation will be stochastic, so we need to force program to only use CPU with single thread.

* training sample shuffling:

During the training, the dataset shuffling will be another part introducing randomness.

Here we target the training code: “model.fit(shuffle=True)”

example:

with tf.Graph().set\_default():

np.random.seed(<seed>) # set random seed for shuffling

tf.set\_random\_seed(<seed>) # set random seed for shuffling

config = tf.ConfigProto(device\_count={“GPU”: 0}, # force to use CPU

intra\_op\_parallelism\_threads=1, # disable CPU multi-threading

inter\_op\_parallelism\_threads=1)

sess = tf.Session(config=config)

tf.keras.backend.set\_session(sess)

<keras\_model\_building>

<keras\_model\_training>

[train seed model, without dropout, batch\_normalization with one epoch, shuffle=False]

control:

rn.seed

np.random.seed

tf.set\_random\_seed

PYTHONHASHSEED

not using GPU

config session

* Can I train two models once and still get two same model?

🡺 success

* which seed control is necessary?

tf.set\_random\_seed

* where can I set seed?
  + top of the code: seccess
* do I need to consistently set seed?

No

* only set seed at the very file top:   
  (rn.seed, np.random.seed, tf.set\_random\_seed, PYTHONHASHSEED)

🡺 success

[train seed model, without dropout, batch\_normalization with one epoch, shuffle=True]

compare to previous experiment, we need additionally control “shuffle”

1. need to seed statement before model construct
2. add set seed statement before model.fit
   * necessary seed control:

np.random\_seed, tf.set\_random\_seed

* + the order of seed control statement doesn’t matter

[train non\_seed model, without dropout, batch\_normalization with one epoch, shuffle=True]

compare to previous experiment, we need a additional control “non\_seed\_model”